Main\_application.py

The first route checks to see if the user is logged in. If so, they are taken to the page where they can decide to logout or search for a book. If the user is not logged in, they are taken to the login page.

The /login route first checks to see if the user is logged in. If so, they are taken to the page where they can search for books. If they are not logged in, the user inputs a username and password. This password is compared to those stored in the database. If there is a match the user is taken to the page where they can search for books. If there isn’t a match, the user is taken to create an account.

The /create route first checks to see if the user is logged in. If so, they are taken to the page where they can search for books. Otherwise, the inputted username and password are stored and compared to the usernames in the database. If there is a match a message appears saying that the username already exists. If not, then the account is created and the user is taken to the login page.

The logout route removes the username from the session and then returns the user to the login page where a message appears stating that the user is logged out.

The /books route first checks to see if the user is logged in. If so, the user inputted value is compared against the books in the database. If there is no match, the user will see a message stating that no such book is in the database. If there is a match, the book will be stored in session and a get request will be made for the book’s average rating and number of reviews. Try and except were used to either return Null or the desired values. Then the user will be taken to the book page with the results displayed. If the user is not logged in, they will be taken to the login page.

The /book route first checks to see if the user is logged in. If so, the book is retrieved from session. Then a get requestion is made to retrieve the amount and average reviews using try and except (like in /books). Then, a search is made in the database to see if the user has already made a review. If they haven’t, a new session is made for the review and the inputted review is added to the review table in the database. Then the review is returned to the book page. If the user has already made a review, a message will appear that says they have already done so.

The /api/isbn route takes the inputted isbn number and compares it to the books database. If there is a match, then the try and except is used to return the required JSON info or NULL. If there is no match in the database, a 404 error is returned.

Import.py

Inside this file, the books, username\_passwords, and the reviews tables are created. In addition, the .csv document containing the book data is imported into the books table.

Tables.py

This page creates the skeletons for the tables using classes that can be called in the routes from the main\_application. Each table has an id that’s set as the primary key. The remaining columns in the tables are set as either strings or integers.

Layout.html

This layout file creates two textboxes, one for the header and one for the body. This is also where the stylesheet and bootstrap are linked.

Layout2.html

This is similar to the other layout file; however, only the login html file uses it. This is because the login page is where the page description is located and thus it will have a slightly different textbox.

Login.html

This uses the /login route and contains two input boxes for the username and password. It also creates a submit button once the user has entered the information.

Logout.html

This uses the /logout route and has a submit button for the user to press in order to logout.

Main1.html

This uses the /books route with a submit button to take the user to the books.html page. This page also uses the /logout route with a submit button to take the user to the /logout page.

Book.html

First, the book information is displayed along with what was gathered from the books API. Then, the /book route is referenced, and the 1-5 input bubbles are created along with a submission button. Then the /books route is used and a button is made to take the user back to the books.html page. At the bottom, the reviews are displayed as text areas.

Books.html

The /books route Is used and an input box is made with a submit button so that the user can submit the inputted information. Then a dropdown menu displays all the corresponding books with a submission button so that the user can be taken to the books.html page. Finally, at the bottom is a link to logout.

Create.html

The /create route is referenced and two input boxes are created for the user to input their username and password, along with a submission button. A login link is also made.